

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月8日 投稿者: SATOXITS

長距離UDPで驚速データ転送を実現

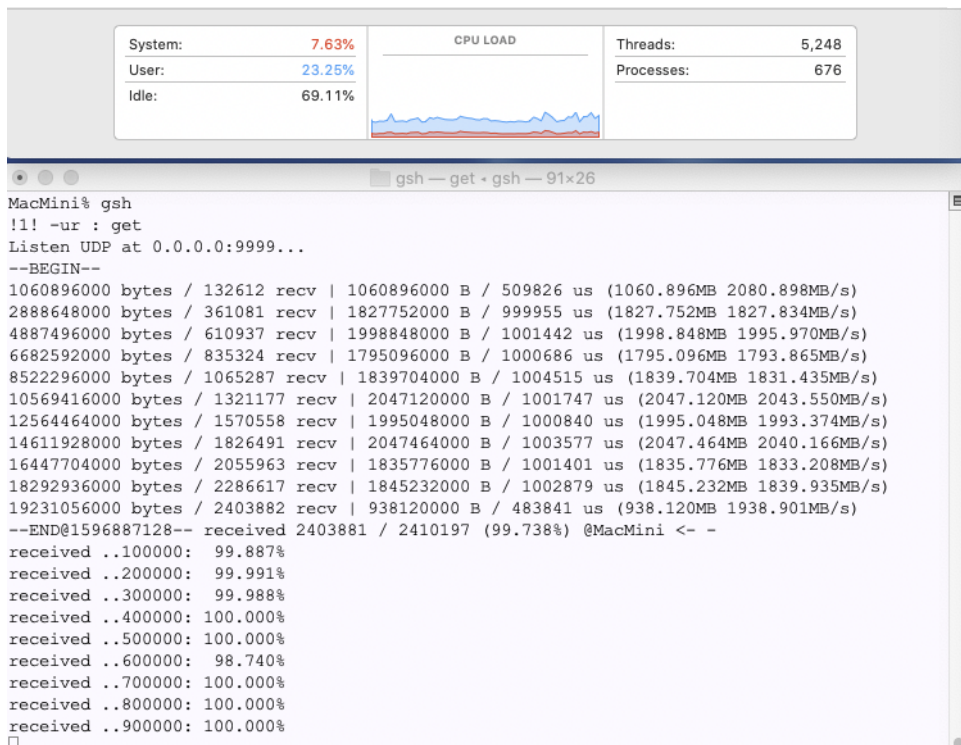
開発：そもそもshellを作ろうと思いついたのは昨日の朝、UDPでならどの程度のデータ転送速度が実現できるか興味を持ったためです。でついつい、Cでゼロからプログラムを書き始めてしまったのですが、しばらくして、いやいやこんな力仕事の書き方はもう卒業すべきだと思い、Goに立ち返ったのでした。

開発：で、ソケットを作ったり繋いだりするのGoのパッケージを使えば簡単に書けるわけです。まあ、ライブラリがあればどんな言語だって簡単ですけど。しかも、Goで使うべき関数を間違えてしばらく戸惑いました。システム関係のパッケージのドキュメントはどれも完成度が低い気がします。まあともかく、やってみた結果がこれです。



基盤：2.27GB/s！

開発：もちろんUDPですから、データは取りこぼすわけです。ですが、同一マシン内で、マシンの負荷が軽ければ、ほぼ受け取れます。この例では 99.7%は受け取れています。



基盤：このMacMiniがこれまで生きてきて一番高速なネットワーク転送です。

社長：使用しているメモリ2.7GHzですからね。まあおそらく64bit、8バイト/クロックでアクセスできるんでしょうけど。いや、32ビットかな？



Memory Slots:

ECC: Disabled
Upgradeable Memory: Yes

BANK 0/ChannelA-DIMM0:

Size: 4 GB
Type: DDR4
Speed: 2667 MHz
Status: OK
Manufacturer: Micron
Part Number: 4ATF51264HZ-2G6E3
Serial Number: 25662DAD

開発：32ビット4バイトだったとすると4 x 2.7 で 10.8GB/s、64ビット8バイトだとすると21.6GB/s は出るはずですね。

基盤：172.8Gbps。

社長：メモリのコピーでは読んで書くから、その半分になる。だからメモリのコピーは最速で10GB/sと思います。

開発：ただ、CPUの命令ループでコピーする場合、1ワードのコピーにCPUクロックが複数かかりますね。ソースとディスティネーションのポインタレジスタの更新、終了判定、

あとはロードとストアと、結局8バイトのコピーに8クロックくらいかかると言うんです。簡単なテストプログラムを書いて測りました。volatile にした 100MBのバッファの間をコピーを10回繰り返すというものです。

```
MacMini% dog -membench
write 1.05GB 0.22s ( 4.81GB/s) - by memset
write 1.05GB 0.09s (11.19GB/s) - by int64 loop
write 1.05GB 0.11s ( 9.53GB/s) - by int32 loop
write 1.05GB 0.33s ( 3.13GB/s) - by int8 loop
copy 1.05GB 0.06s (18.19GB/s) - by memcpy
copy 1.05GB 0.14s ( 7.45GB/s) - by int64 loop
copy 1.05GB 0.15s ( 6.79GB/s) - by int32 loop
copy 1.05GB 0.62s ( 1.70GB/s) - by int8 loop
```

基盤：理論値の10GB/s出てますね。

社長：ユーザからDMAは使えないんですかね？

開発：さあ。このmemcpyの値は、異常に速かったり遅かったりします。デュアルポートメモリでも無いでしょうに18GB/s出ているのも謎です。他のCPUループでの実装はほぼ安定していますので、ひょっとしてDMAを使っているのかも知れません。デバイスとして使えるのでしょうか。システムコールで叩いてるのでしょうか？

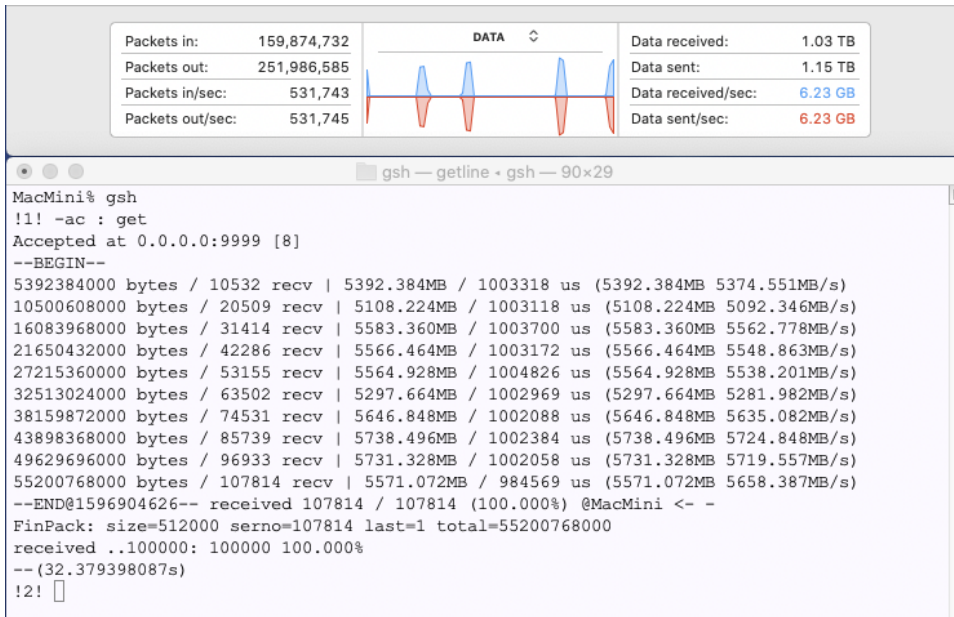
基盤：通信経由のコピーがプロセス内でのメモリーメモリコピーと同等というのはびっくりですね。

開発：それも実際にはDMAによるメモリーメモリコピーに帰着させてるんじゃないかとは思いますがね。他のマシンへの転送では、数マイクロ秒間隔を開けてやらないと、OSのバッファがいっぱいになってエラーが出るのですが、同一マシン内だとそれが出ないんです。

社長：自分自身に送信してコピーすると高速化したりするかもですねw

* * *

開発：それで、次はTCPではどうなのかといことになります。やってみた結果がこれ。



基盤：ネットが6.23Gbpsに到達しております！

開発：このMacMiniは、外に対しては1GB/sでしか繋がりませんが、内部的には違う話ですね。プロセス間での通信は、プロセス内部でのメモリのコピーと同等の性能が出ていると見えます。

社長：TCPのほうが速いんですね。

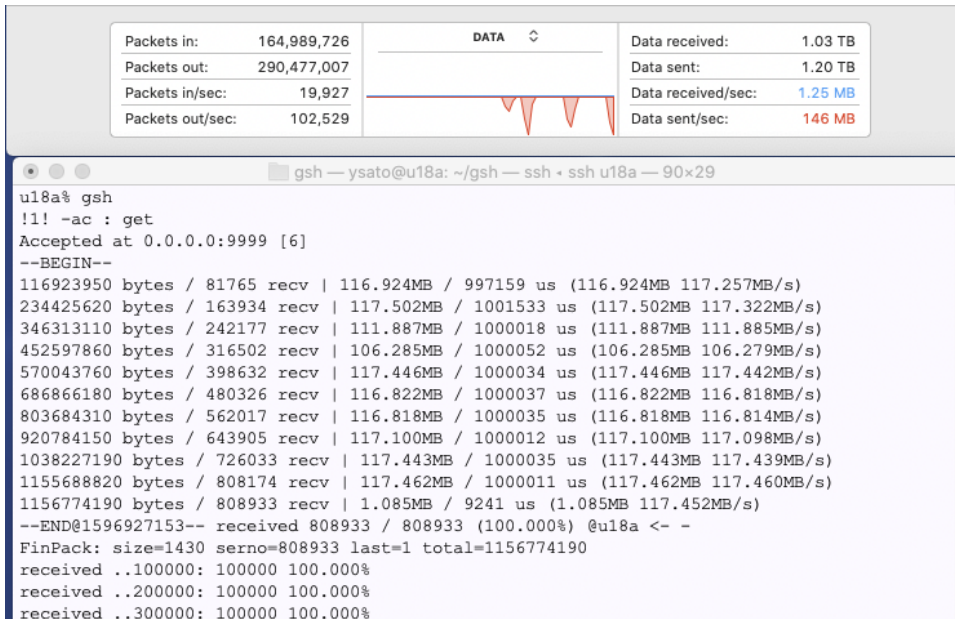
開発：UDPだと固定長パケットで、8KBを超えると起こられてしまうわけです。だから送信の回数、OSとの通信が多くなります。一方TCPではそういう制約は無いですから、でっかいバッファのありかをどーんとOSに教えてやって任せればよいわけです。200KB程度のサイズまでは、サイズが大きいほど速くなります。

基盤：ただこれは、同一マシン内での通信の特殊状況なんでしょうね。

開発：そうですね。ping への RTT が 0.05ms とかいう世界です。これが、隣のマシンになると 0.5ms程度はどうしてもかかる。さらにWANに出ると速くても5ms程度はかかる。アジア近隣で50ms、アメリカまで行くと130ms、ヨーロッパまで行くと250msになってしまうわけです。相手の受信確認を頻繁に行うTCPではこれは、まるで別の世界になってしまいます。

* * *

開発：されそれで、RTT 0.5ms程度の隣のマシンとの転送ですが、sendのサイズは1500バイトで十分に1Gbpsを使い切ることがわかりました。



開発：一方UDPの成績は芳しくありません。どうも性能が出ない…

* * *

開発：えーと、調整をした結果、まずまずの性能に落ち着きました。

```

gsh — ysato@u18a: ~/gsh — ssh • ssh u18a — 71x43
u18a% gsh
!! -iu : get
Listen UDP at 0.0.0.0:9999...
1596943433.762 --BEGIN--
+1.000 108.139460MB / 75622 rcv ||| 108.139MB / 1.000s (108.128MB/s)
+2.000 215.458100MB / 150670 rcv ||| 107.319MB / 1.000s (107.319MB/s)
+3.000 328.210740MB / 229518 rcv ||| 112.753MB / 1.000s (112.748MB/s)
+4.000 439.391810MB / 307267 rcv ||| 111.181MB / 1.000s (111.174MB/s)
+5.000 544.010610MB / 380427 rcv ||| 104.619MB / 1.000s (104.606MB/s)
+6.000 656.171230MB / 458861 rcv ||| 112.161MB / 1.000s (112.156MB/s)
+7.000 769.156960MB / 537872 rcv ||| 112.986MB / 1.000s (112.984MB/s)
+8.000 882.445850MB / 617095 rcv ||| 113.289MB / 1.000s (113.285MB/s)
+9.000 994.443450MB / 695415 rcv ||| 111.998MB / 1.000s (111.997MB/s)
+9.992 1097.909670MB / 767769 rcv ||| 103.466MB / 0.991s (104.355MB/s)
1596943443.754 --END-- received 767769 / 767769 (100.000%) @u18a <- -
FinPack: packsize=1430, serno=767769, last=1, total=1097909670
Total: 9.992s, 1097.909670MB, 104.789MiB/s (879.034Mbps 838.312Mibps)
received ..100000: 100000 100.000%
received ..200000: 100000 100.000%
received ..300000: 100000 100.000%
received ..400000: 100000 100.000%
received ..500000: 100000 100.000%
received ..600000: 100000 100.000%
received ..700000: 100000 100.000%
--(45.155099971s)
!2! -it : get
Accepted TCP at 0.0.0.0:9999 [6]
1596943466.280 --BEGIN--
+1.000 117.964800MB / 225 rcv ||| 117.965MB / 1.000s (117.957MB/s)
+2.000 235.405312MB / 449 rcv ||| 117.441MB / 1.000s (117.441MB/s)
+3.000 352.845824MB / 673 rcv ||| 117.441MB / 1.000s (117.438MB/s)
+4.000 470.286336MB / 897 rcv ||| 117.441MB / 1.000s (117.439MB/s)
+5.000 587.726848MB / 1121 rcv ||| 117.441MB / 1.000s (117.437MB/s)
+6.000 705.691648MB / 1346 rcv ||| 117.965MB / 1.000s (117.958MB/s)
+7.000 823.132160MB / 1570 rcv ||| 117.441MB / 1.000s (117.439MB/s)
+8.000 940.572672MB / 1794 rcv ||| 117.441MB / 1.000s (117.439MB/s)
+9.000 1058.013184MB / 2018 rcv ||| 117.441MB / 1.000s (117.439MB/s)
+9.997 1175.453696MB / 2242 rcv ||| 117.441MB / 0.997s (117.838MB/s)
1596943476.277 --END-- received 2242 / 2242 (100.000%) @u18a <- -
FinPack: packsize=524288, serno=2242, last=1, total=1175453696
Total: 9.997s, 1175.453696MB, 112.135MiB/s (940.657Mbps 897.080Mibps)
--(19.609383142s)
!3!

```

受信側

```

gsh — getline • gsh — 71x32
MacMini% gsh
!!! -ou u18a:9999 gen
Socket: connected to u18a:9999 [192.168.10.18:9999], socket[9]
-- interval = 4us, size = 1430B
+1.000 108.148040MB / 75628 sent ||| 108.148MB / 1.000s (108.136MB/s)
+2.000 215.456670MB / 150669 sent ||| 107.309MB / 1.000s (107.285MB/s)
+3.002 328.373760MB / 229632 sent ||| 112.917MB / 1.001s (112.756MB/s)
+4.003 439.604880MB / 307416 sent ||| 111.231MB / 1.001s (111.152MB/s)
+5.004 544.348090MB / 380663 sent ||| 104.743MB / 1.001s (104.594MB/s)
+6.004 656.605950MB / 459165 sent ||| 112.258MB / 1.000s (112.224MB/s)
+7.006 769.746120MB / 538284 sent ||| 113.140MB / 1.001s (112.993MB/s)
+8.006 883.100790MB / 617553 sent ||| 113.355MB / 1.000s (113.309MB/s)
+9.006 995.148440MB / 695908 sent ||| 112.048MB / 1.000s (112.017MB/s)
-- Sent FinPack: dsize=1097909670
+9.992 1097.909670MB / 767769 sent ||| 102.761MB / 0.986s (104.264MB/s)
--(10.010178932s)
!2! -ot u18a:9999 gen
Socket: connected to u18a:9999, socket[9]
-- interval = 0us, size = 524288B
+1.000 119.013376MB / 227 sent ||| 119.013MB / 0.993s (119.795MB/s)
+2.000 236.453888MB / 451 sent ||| 117.441MB / 1.000s (117.418MB/s)
+3.000 353.894400MB / 675 sent ||| 117.441MB / 1.000s (117.427MB/s)
+4.001 471.859200MB / 900 sent ||| 117.965MB / 1.000s (117.945MB/s)
+5.001 589.299712MB / 1124 sent ||| 117.441MB / 1.000s (117.421MB/s)
+6.001 706.740224MB / 1348 sent ||| 117.441MB / 1.000s (117.415MB/s)
+7.001 824.180736MB / 1572 sent ||| 117.441MB / 1.000s (117.415MB/s)
+8.001 941.621248MB / 1796 sent ||| 117.441MB / 1.000s (117.417MB/s)
+9.002 1059.586048MB / 2021 sent ||| 117.965MB / 1.000s (117.950MB/s)
-- Sent FinPack: dsize=1175453696
+9.987 1175.453696MB / 2242 sent ||| 115.868MB / 0.985s (117.634MB/s)
--(9.994051152s)
!3!

```

送信側

開発：ペイロードぶんのスピードとして TCPでは 940Mbps出ていますので、理論上の最高性能かなと思います。一方UDPでは 880Mbps程度。連続して送るとシステムのバッファが溢れてしまうので、4マイクロ間隔で1430Bを送っています。これだと、バッファがいっぱいだよエラーが出ませんし、受け側もほぼ100%受け取れます。これ以上間隔を短くすると、送り側としてはTCP並の速度で送れますが、受け側がボロボロになります。

開発：ああ、でも3usなら送信側が再試行すれば100%受け取られますね。

```

!3! -iu : get
Listen UDP at 0.0.0.0:9999...
1596944605.890 --BEGIN--
+1.000 115.951550MB / 81085 recv ||| 115.952MB / 1.000s (115.931MB/s)
+2.000 233.704900MB / 163430 recv ||| 117.753MB / 1.000s (117.749MB/s)
+3.000 344.283940MB / 240758 recv ||| 110.579MB / 1.000s (110.572MB/s)
+4.000 461.910020MB / 323014 recv ||| 117.626MB / 1.000s (117.621MB/s)
+5.000 577.954520MB / 404164 recv ||| 116.044MB / 1.000s (116.036MB/s)
+6.000 696.359950MB / 486965 recv ||| 118.405MB / 1.000s (118.396MB/s)
+7.000 815.477520MB / 570264 recv ||| 119.118MB / 1.000s (119.114MB/s)
+8.001 926.002220MB / 647554 recv ||| 110.525MB / 1.000s (110.514MB/s)
+9.001 1044.263220MB / 730254 recv ||| 118.261MB / 1.000s (118.257MB/s)
+9.993 1161.049890MB / 811923 recv ||| 116.787MB / 0.993s (117.655MB/s)
1596944615.883 --END-- received 811923 / 811923 (100.000%) @u18a <- -
FinPack: packsize=1430, serno=811923, last=1, total=1161049890
Total: 9.993s, 1161.049890MB, 110.801MiB/s (929.466Mbps 886.407Mibps)
received ..100000: 100000 100.000%
received ..200000: 100000 100.000%
received ..300000: 100000 100.000%
received ..400000: 100000 100.000%
received ..500000: 100000 100.000%
received ..600000: 100000 100.000%
received ..700000: 100000 100.000%
received ..800000: 100000 100.000%
-- (33.730971752s)

```

3us間隔送信

```

!4! -iu : get
Listen UDP at 0.0.0.0:9999...
1596944663.886 --BEGIN--
+1.000 117.912080MB / 82456 recv ||| 117.912MB / 1.000s (117.897MB/s)
+2.000 234.894660MB / 164262 recv ||| 116.983MB / 1.000s (116.974MB/s)
+3.000 352.582230MB / 246561 recv ||| 117.688MB / 1.000s (117.680MB/s)
+4.000 469.979510MB / 328657 recv ||| 117.397MB / 1.000s (117.395MB/s)
+5.000 585.163150MB / 409205 recv ||| 115.184MB / 1.000s (115.165MB/s)
+6.001 702.856440MB / 491508 recv ||| 117.693MB / 1.000s (117.675MB/s)
+7.001 819.734630MB / 573241 recv ||| 116.878MB / 1.000s (116.875MB/s)
+8.001 936.607100MB / 654970 recv ||| 116.872MB / 1.000s (116.856MB/s)
+9.001 1054.218880MB / 737216 recv ||| 117.612MB / 1.000s (117.604MB/s)
+10.001 1170.191880MB / 818316 recv ||| 115.973MB / 1.000s (115.972MB/s)
+10.009 1171.170000MB / 819000 recv ||| 0.978MB / 0.008s (120.473MB/s)
1596944673.895 --END-- received 819000 / 849585 (96.400%) @u18a <- -
FinPack: packsize=1430, serno=849585, last=1, total=1214906550
Total: 10.009s, 1171.170000MB, 111.591MiB/s (936.094Mbps 892.729Mibps)
received ..100000: 96552 96.552%
received ..200000: 94711 94.711%
received ..300000: 96268 96.268%
received ..400000: 98731 98.731%
received ..500000: 97182 97.182%
received ..600000: 94692 94.692%
received ..700000: 95912 95.912%
received ..800000: 96843 96.843%
-- (13.789123487s)

```

2us間隔送信


```

!5! -iu : get
Listen UDP at 0.0.0.0:9999...
1596944704.402 --BEGIN--
+1.000 118.103700MB / 82590 recv ||| 118.104MB / 1.000s (118.068MB/s)
+2.000 235.559610MB / 164727 recv ||| 117.456MB / 1.000s (117.463MB/s)
+3.000 352.811030MB / 246721 recv ||| 117.251MB / 1.000s (117.238MB/s)
+4.001 470.038140MB / 328698 recv ||| 117.227MB / 1.001s (117.143MB/s)
+5.001 584.971530MB / 409071 recv ||| 114.933MB / 1.000s (114.928MB/s)
+6.001 702.633360MB / 491352 recv ||| 117.662MB / 1.000s (117.656MB/s)
+7.001 820.818570MB / 573999 recv ||| 118.185MB / 1.000s (118.175MB/s)
+8.001 938.248740MB / 656118 recv ||| 117.430MB / 1.000s (117.424MB/s)
+9.001 1054.999660MB / 737762 recv ||| 116.751MB / 1.000s (116.738MB/s)
+9.990 1169.615590MB / 817913 recv ||| 114.616MB / 0.989s (115.922MB/s)
1596944714.392 --END-- received 817913 / 887332 (92.177%) @u18a <- -
FinPack: packsize=1430, serno=887332, last=1, total=1268884760
Total: 9.990s, 1169.615590MB, 111.653MiB/s (936.611Mbps 893.222Mibps)
received ..100000: 91622 91.622%
received ..200000: 91200 91.200%
received ..300000: 90051 90.051%
received ..400000: 95690 95.690%
received ..500000: 92985 92.985%
received ..600000: 91323 91.323%
received ..700000: 89582 89.582%
received ..800000: 95160 95.160%
--(13.365294508s)

```

1us間隔送信

```

!6! -iu : get
Listen UDP at 0.0.0.0:9999...
1596944736.783 --BEGIN--
+1.001 108.760080MB / 76056 recv ||| 108.760MB / 1.001s (108.600MB/s)
+2.001 225.922840MB / 157988 recv ||| 117.163MB / 1.000s (117.161MB/s)
+3.002 342.244760MB / 239332 recv ||| 116.322MB / 1.000s (116.306MB/s)
+4.002 459.769310MB / 321517 recv ||| 117.525MB / 1.000s (117.523MB/s)
+5.002 577.272410MB / 403687 recv ||| 117.503MB / 1.000s (117.500MB/s)
+6.002 691.159040MB / 483328 recv ||| 113.887MB / 1.000s (113.885MB/s)
+7.002 754.210600MB / 527420 recv ||| 63.052MB / 1.000s (63.041MB/s)
+8.002 866.867430MB / 606201 recv ||| 112.657MB / 1.000s (112.645MB/s)
+9.002 971.830860MB / 679602 recv ||| 104.963MB / 1.000s (104.957MB/s)
+10.002 1089.694320MB / 762024 recv ||| 117.863MB / 1.000s (117.860MB/s)
+10.009 1090.525150MB / 762605 recv ||| 0.831MB / 0.007s (119.682MB/s)
1596944746.792 --END-- received 762605 / 1551520 (49.152%) @u18a <- -
FinPack: packsize=1430, serno=1551520, last=1, total=2218673600
Total: 10.009s, 1090.525150MB, 103.907MiB/s (871.635Mbps 831.256Mibps)
received ..100000: 40691 40.691%
received ..200000: 40179 40.179%
received ..300000: 44043 44.043%
received ..400000: 42903 42.903%
received ..500000: 39650 39.650%
received ..600000: 39105 39.105%
received ..700000: 41170 41.170%
received ..800000: 39558 39.558%
received ..900000: 40063 40.063%

```

無間隔送信

基盤：ぼろぼろですねw

社長：要するにLANというかレイテンシーがサブミリ秒の近接したノード間でUDPを使うメリットは無いということですね。

開発：そう思います。で、問題はレイテンシーが大きい時はどうか、ということで、本題になります。なお、この実験は電力を約60W食うことが観測されています（笑）

* * *

開発：まあこれは実は最初にやって結果はわかっただけで、比較のためにマシン内や近接マシンで試すのに時間をとられてしまいました。

開発：ではまずは最果ての地、ドイツフランクフルト支所。RTT 250ms 超です。ここは scp のアップロードで 1MB/s しか出ないことがわかっています。このベンチマークでも TCP でのデータ送信は同様な結果になります。

```

gsh — ysato@ip-172-26-1-22:~/gsh — ssh -i ~/.ssh/de1.pem de1 — 89x22
de1$
de1$
de1$
de1$
de1$ gsh
!! -it : get
Accepted TCP at 0.0.0.0:9999 [6]
1596945880.624 --BEGIN--
+1.000 2.621440MB / 5 recv ||| 2.621MB / 1.000s (2.621MB/s)
+2.000 4.194304MB / 8 recv ||| 1.573MB / 1.000s (1.573MB/s)
+3.000 5.767168MB / 11 recv ||| 1.573MB / 1.000s (1.573MB/s)
+4.000 8.912896MB / 17 recv ||| 3.146MB / 1.000s (3.145MB/s)
+5.001 9.961472MB / 19 recv ||| 1.049MB / 1.000s (1.048MB/s)
+6.001 11.010048MB / 21 recv ||| 1.049MB / 1.000s (1.048MB/s)
+10.001 11.534336MB / 22 recv ||| 0.524MB / 4.001s (0.131MB/s)
+12.001 12.058624MB / 23 recv ||| 0.524MB / 2.000s (0.262MB/s)
+12.099 12.533888MB / 24 recv ||| 0.475MB / 0.098s (4.845MB/s)
Ma 1596945892.724 --END-- received 24 / 24 (100.000%) @ip-172-26-1-22 <- -
ph FinPack: packsize=524288, serno=24, last=1, total=12582912
ss Total: 12.100s, 12.533888MB, 0.988MiB/s (8.287Mbps 7.903Mibps)
Ar -- (34.597390122s)
On !2!

```

Packets in:	169,060,128	DATA	Data received:	1.03 TB
Packets out:	400,808,484		Data sent:	1.36 TB
Packets in/sec:	86		Data received/sec:	7.02 KB
Packets out/sec:	166		Data sent/sec:	228 KB

```

gsh — gettime - gsh — 90x16
MacMini%
MacMini% gsh
!! -ot del:9999 gen
Socket: connected to del:9999, socket[9]
-- interval = 0us, size = 524288B
+2.007 0.524288MB / 1 sent ||| 0.524MB / 0.037s (14.176MB/s)
+3.008 2.621440MB / 5 sent ||| 2.097MB / 1.000s (2.096MB/s)
+4.013 4.718592MB / 9 sent ||| 2.097MB / 1.005s (2.087MB/s)
+5.018 5.242880MB / 10 sent ||| 0.524MB / 1.005s (0.522MB/s)
+6.019 9.437184MB / 18 sent ||| 4.194MB / 1.002s (4.188MB/s)
+7.021 11.534336MB / 22 sent ||| 2.097MB / 1.001s (2.095MB/s)
+14.040 12.058624MB / 23 sent ||| 0.524MB / 7.020s (0.075MB/s)
-- Sent FinPack: dsize=12582912
+14.040 12.533888MB / 24 sent ||| 0.475MB / 0.000s (23763.200MB/s)
-- (14.309271159s)
!2!

```

基盤：しかもスピードが時々刻々かわっている。ネットの状況に左右されてるんでしょうね。遠隔との通信の速度は、非常に不安定であることは実感としてわかっています。

開発：そしていよいよ、UDPではどうか。

The screenshot shows a terminal window with the following output:

```

!3! -iu : get
Listen UDP at 0.0.0.0:9999...
1596946196.483 --BEGIN--
+1.000 62.684050MB / 43835 recv ||| 62.684MB / 1.000s (62.664MB/s)
+2.000 125.789950MB / 87965 recv ||| 63.106MB / 1.000s (63.115MB/s)
+3.000 188.482580MB / 131806 recv ||| 62.693MB / 1.000s (62.688MB/s)
+4.000 248.592630MB / 173841 recv ||| 60.110MB / 1.000s (60.106MB/s)
+5.000 297.475750MB / 208025 recv ||| 48.883MB / 1.000s (48.879MB/s)
+6.001 345.004660MB / 241262 recv ||| 47.529MB / 1.000s (47.524MB/s)
+7.001 397.048080MB / 277656 recv ||| 52.043MB / 1.000s (52.039MB/s)
+8.001 449.194460MB / 314122 recv ||| 52.146MB / 1.000s (52.142MB/s)
+9.001 498.083300MB / 348310 recv ||| 48.889MB / 1.000s (48.885MB/s)
+10.001 545.118860MB / 381202 recv ||| 47.036MB / 1.000s (47.032MB/s)
+10.008 545.120290MB / 381203 recv ||| 0.001MB / 0.007s (0.199MB/s)
1596946206.491 --END-- received 381203 / 755165 (50.479%) @ip-172-26-1-22 <- -
FinPack: packsize=1430, serno=755165, last=1, total=1079885950
Total: 10.008s, 545.120290MB, 51.945MiB/s (435.744Mbps 415.558Mibps)
received ..100000: 58231 58.231%
received ..200000: 64057 64.057%
received ..300000: 54461 54.461%
received ..400000: 44359 44.359%
received ..500000: 44499 44.499%
received ..600000: 46506 46.506%
received ..700000: 43648 43.648%
--(11.999824580s)
!4!

```

Below the terminal is a network monitoring tool interface showing:

Packets in:	169,064,187	DATA	Data received:	1.03 TB
Packets out:	404,538,483		Data sent:	1.36 TB
Packets in/sec:	8		Data received/sec:	1.53 KB
Packets out/sec:	81,471		Data sent/sec:	114 MB

At the bottom, another terminal window shows the following output:

```

--(10.025925725s)
!3! -ou del:9999 gen
Socket: connected to del:9999 [52.59.53.30:9999], socket[9]
-- interval = 4us, size = 1430B
+1.005 108.782960MB / 76072 sent ||| 108.783MB / 1.005s (108.282MB/s)
+2.006 208.361010MB / 145707 sent ||| 99.578MB / 1.001s (99.472MB/s)
+3.006 312.450710MB / 218497 sent ||| 104.090MB / 1.000s (104.085MB/s)
+4.006 422.044480MB / 295136 sent ||| 109.594MB / 1.000s (109.574MB/s)
+5.008 535.555020MB / 374514 sent ||| 113.511MB / 1.002s (113.277MB/s)
+6.009 647.047830MB / 452481 sent ||| 111.493MB / 1.001s (111.401MB/s)
+7.010 757.602560MB / 529792 sent ||| 110.555MB / 1.001s (110.477MB/s)
+8.010 864.100380MB / 604266 sent ||| 106.498MB / 1.001s (106.417MB/s)
+9.015 976.931670MB / 683169 sent ||| 112.831MB / 1.005s (112.278MB/s)
+10.016 1079.884520MB / 755164 sent ||| 102.953MB / 1.001s (102.877MB/s)
-- Sent FinPack: dsize=1079885950
+10.016 1079.885950MB / 755165 sent ||| 0.001MB / 0.000s (18.333MB/s)
--(10.022592687s)
!4!

```

基盤：50MB/s で届いていますね！**50倍速**！

開発：まあ、半分落ちてますけどね。

社長：といたしますか、我社から社外に向けて実際にギガビットでデータを送出したのが初めてですね。

開発：それでこの数値は、相手がドイツでも東京でも大差無いんですよ。つまり、うちのインターネットは、上りは500Mbpsしか出ないのではないかという気がします。

社長：ライトセール以外の相手にためしてみたいところですが。CPU負荷的に厳しいとか。

開発：全般的には余裕ですね。30%食うことはないです。なにせデータを受けるだけで何も処理してませんし。

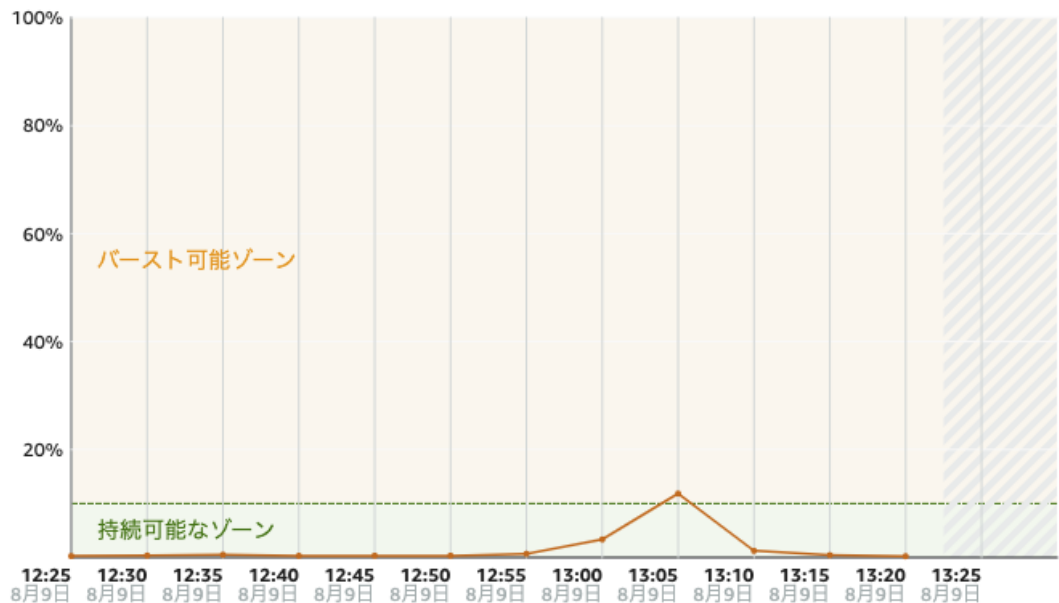
```

ysato — ysato@ip-172-26-1-22:~ — ssh - ssh -i ~/.ssh/de1.pem de1 — 76x13
top - 13:34:40 up 19 days, 23:40, 4 users, load average: 0.00, 0.00, 0.00
Tasks: 111 total, 3 running, 82 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.6%sy, 0.0%ni, 88.8%id, 0.0%wa, 0.0%hi, 0.6%si, 10.1
Mem: 1009136k total, 830448k used, 178688k free, 111288k buffers
Swap: 0k total, 0k used, 0k free, 583108k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 20209 ysato    20   0  205m 1728  888  R  28.6   0.2   0:01.95  get
  2326 root      20   0   6524   96    0  R   0.7   0.0   0:33.34  rngd
    1 root      20   0 19696 2592 2264  S   0.0   0.3   0:02.27  init
    2 root      20   0    0    0    0  S   0.0   0.0   0:00.00  kthreadd
    4 root      0  -20    0    0    0  I   0.0   0.0   0:00.00  kworker/0:0H
    6 root      0  -20    0    0    0  I   0.0   0.0   0:00.00  mm_percpu_wq

  gsh — ysato@ip-172-26-1-22:~/gsh — ssh -i ~/.ssh/de1.pem de1 — 76x20
Total: 10.020s, 600.877420MB, 57.192MiB/s (479.761Mbps 457.535Mibps)
received ..100000: 57337 57.337%
received ..200000: 54751 54.751%
received ..300000: 57050 57.050%
received ..400000: 56637 56.637%
received ..500000: 58305 58.305%
received ..600000: 48783 48.783%
received ..700000: 55565 55.565%
--(38.541960564s)
!?! -iu : get
Listen UDP at 0.0.0.0:9999...
1596947673.635 --BEGIN--
st +1.000 62.642580MB / 43806 recv ||| 62.643MB / 1.000s (62.624MB/s)
riv +2.000 122.220670MB / 85469 recv ||| 59.578MB / 1.000s (59.585MB/s)
in +3.000 184.387060MB / 128942 recv ||| 62.166MB / 1.000s (62.161MB/s)
Ma +4.000 248.014910MB / 173437 recv ||| 63.628MB / 1.000s (63.623MB/s)
ih +5.000 307.807500MB / 215250 recv ||| 59.793MB / 1.000s (59.788MB/s)
st +6.000 371.465380MB / 259766 recv ||| 63.658MB / 1.000s (63.653MB/s)
ut +7.001 434.232370MB / 303659 recv ||| 62.767MB / 1.000s (62.761MB/s)
  
```

5分あたりの平均 CPU 使用率



開発：たとえばライトセール東京にある我社ネームサーバ。

The image shows a terminal window with the following output:

```

jpl$ gsh
!! -iu : get
Listen UDP at 0.0.0.0:9999...
1596946912.678 --BEGIN--
+1.000 61.142510MB / 42757 recv ||| 61.143MB / 1.000s (61.127MB/s)
+2.000 123.138730MB / 86111 recv ||| 61.996MB / 1.000s (62.001MB/s)
+3.000 186.775160MB / 130612 recv ||| 63.636MB / 1.000s (63.632MB/s)
+4.000 249.386280MB / 174396 recv ||| 62.611MB / 1.000s (62.606MB/s)
+5.000 309.227490MB / 216243 recv ||| 59.841MB / 1.000s (59.832MB/s)
+6.001 371.542600MB / 259820 recv ||| 62.315MB / 1.000s (62.310MB/s)
+7.001 434.927350MB / 304145 recv ||| 63.385MB / 1.000s (63.381MB/s)
+8.001 482.214590MB / 337213 recv ||| 47.287MB / 1.000s (47.283MB/s)
+9.001 527.385430MB / 368801 recv ||| 45.171MB / 1.000s (45.167MB/s)
+10.001 578.211920MB / 404344 recv ||| 50.826MB / 1.000s (50.823MB/s)
+10.017 578.452160MB / 404512 recv ||| 0.240MB / 0.016s (15.001MB/s)
1596946922.695 --END-- received 404512 / 736993 (54.887%) @ip-172-26-1-84 <- -
FinPack: packsize=1430, serno=736993, last=1, total=1053899990
Total: 10.017s, 578.452160MB, 55.072MiB/s (461.979Mbps 440.578Mibps)
received ..100000: 64640 64.640%
received ..200000: 60287 60.287%
received ..300000: 56945 56.945%
received ..400000: 58995 58.995%
received ..500000: 60064 60.064%
received ..600000: 43841 43.841%
received ..700000: 43008 43.008%
--(20.273957885s)
!2!

```

Below the terminal is a network monitoring tool showing traffic between two hosts:

Packets in:	169,071,367	DATA	Data received:	1.03 TB
Packets out:	405,346,239		Data sent:	1.36 TB
Packets in/sec:	11		Data received/sec:	1.88 KB
Packets out/sec:	13		Data sent/sec:	2.08 KB

Below the network tool is another terminal window showing UDP data transmission statistics:

```

--(10.022592687s)
!4! -ou jpl:9999 gen
Socket: connected to jpl:9999 [18.181.38.61:9999], socket[9]
-- interval = 4us, size = 1430B
+1.005 91.663000MB / 64100 sent ||| 91.663MB / 1.005s (91.201MB/s)
+2.007 195.192140MB / 136498 sent ||| 103.529MB / 1.002s (103.337MB/s)
+3.007 303.793490MB / 212443 sent ||| 108.601MB / 1.000s (108.568MB/s)
+4.008 414.069370MB / 289559 sent ||| 110.276MB / 1.001s (110.184MB/s)
+5.009 515.008780MB / 360146 sent ||| 100.939MB / 1.001s (100.872MB/s)
+6.010 617.355310MB / 431717 sent ||| 102.347MB / 1.001s (102.253MB/s)
+7.012 727.942930MB / 509051 sent ||| 110.588MB / 1.002s (110.371MB/s)
+8.012 837.676840MB / 585788 sent ||| 109.734MB / 1.000s (109.701MB/s)
+9.014 946.569910MB / 661937 sent ||| 108.893MB / 1.002s (108.665MB/s)
+10.017 1053.898560MB / 736992 sent ||| 107.329MB / 1.003s (106.965MB/s)
-- Sent FinPack: dsize=1053899990
+10.018 1053.899990MB / 736993 sent ||| 0.001MB / 0.000s (11.721MB/s)
--(10.024445215s)
!5!

```

開発：もともとパケットがロスするのは織り込み済みですから、到達してないパケットのマップを返してもらって再送すればよいわけですが、ではどのくらいの速度までロスが起らないかをためてみます。… … 6.28MiB/s、50Mbps までは大丈夫っぽいですね。

```

gsh — ysato@ip-172-26-1-22:~/gsh — ssh -i ~/.ssh/de1.pem de1 — 76x
!15! -iu : get
Listen UDP at 0.0.0.0:9999...
1596948253.628 --BEGIN--
+1.000 6.652360MB / 4652 recv ||| 6.652MB / 1.000s (6.651MB/s)
+2.000 13.300430MB / 9301 recv ||| 6.648MB / 1.000s (6.649MB/s)
+3.000 19.974240MB / 13968 recv ||| 6.674MB / 1.000s (6.673MB/s)
+4.000 26.639470MB / 18629 recv ||| 6.665MB / 1.000s (6.665MB/s)
+5.000 32.925750MB / 23025 recv ||| 6.286MB / 1.000s (6.286MB/s)
+6.001 39.590980MB / 27686 recv ||| 6.665MB / 1.000s (6.665MB/s)
+7.001 46.300540MB / 32378 recv ||| 6.710MB / 1.000s (6.709MB/s)
+8.001 52.861380MB / 36966 recv ||| 6.561MB / 1.000s (6.560MB/s)
+9.001 59.556640MB / 41648 recv ||| 6.695MB / 1.000s (6.695MB/s)
+10.001 65.924430MB / 46101 recv ||| 6.368MB / 1.000s (6.367MB/s)
+10.009 65.925860MB / 46102 recv ||| 0.001MB / 0.009s (0.168MB/s)
1596948263.637 --END-- received 46102 / 46102 (100.000%) @ip-172-2
-
FinPack: packsize=1430, serno=46102, last=1, total=65925860
Total: 10.009s, 65.925860MB, 6.281MiB/s (52.691Mbps 50.250Mibps)
--(14.971489287s)
!16! □

gsh — getline · gsh — 77x17
!15! -ou de1:9999 gen 161
Socket: connected to de1:9999 [52.59.53.30:9999], socket[9]
-- interval = 161us, size = 1430B
+1.000 6.646640MB / 4648 sent ||| 6.647MB / 1.000s (6.647MB/s)
+2.004 13.321880MB / 9316 sent ||| 6.675MB / 1.004s (6.646MB/s)
+3.005 20.002840MB / 13988 sent ||| 6.681MB / 1.001s (6.676MB/s)
+4.007 26.678080MB / 18656 sent ||| 6.675MB / 1.002s (6.665MB/s)
+5.007 32.974370MB / 23059 sent ||| 6.296MB / 1.000s (6.293MB/s)
+6.008 39.631020MB / 27714 sent ||| 6.657MB / 1.000s (6.654MB/s)
+7.008 46.344870MB / 32409 sent ||| 6.714MB / 1.000s (6.711MB/s)
+8.008 52.904280MB / 36996 sent ||| 6.559MB / 1.000s (6.559MB/s)
+9.009 59.608120MB / 41684 sent ||| 6.704MB / 1.001s (6.698MB/s)
+10.010 65.924430MB / 46101 sent ||| 6.316MB / 1.001s (6.311MB/s)
-- Sent FinPack: dsize=65925860
+10.010 65.925860MB / 46102 sent ||| 0.001MB / 0.000s (13.241MB/s)
--(10.016077968s)
!16! □

```

開発：161マイクロ秒間隔での送信です。

社長：刻みましたねw

開発：普通のデータは1ビットでも落ちたらいけません、画像とかはとりあえず届いたものから表示するとか、ありますよね。動画だったらなんだか画質が落ちるとか。

社長：これはそもそも、差分だけ送るという方法に相性が良いですね。rsync とはちょっと違う方向性を考えると面白いかも。

基盤：ところで我社の\$5ライトセールは2TB/月という制約ですが、どういうふうにリミットしているんでしょう？もうけっこう上限のような。

開発：ドイツ支部は暇だし死んでも大丈夫ですw でも、テスト用のインスタンスを作ってやってみますかね。

経理：まさか、上限を超えたぶんは従量課金になるとかではないですよね…

— 2020-0809 SatoxITS

